A Taxonomy of Visualization Techniques using the Data State Reference Model

Ed H. Chi

Xerox Palo Alto Research Center

3333 Coyote Hill Road, Palo Alto, CA 94301

echi@parc.xerox.com

ABSTRACT

In previous work, researchers have attempted to construct taxonomies of information visualization techniques by examining the data domains that are compatible with these techniques. This is useful because implementers can quickly identify various techniques that can be applied to their domain of interest. However, these taxonomies do not help the implementers understand how to apply and implement these techniques. In this paper, we will extend and then propose a new way to taxonomize information visualization techniques by using the Data State Model [Chi98]. In fact, as the taxonomic analysis in this paper will show, many of the techniques share similar operating steps that can easily be reused. The paper shows that the Data State Model not only helps researchers understand the space of design, but also helps implementers understand how information visualization techniques can be applied more broadly.

Keywords

Information Visualization, Data State Model, Reference Model, Taxonomy, Techniques, Operators.

1. INTRODUCTION

There have been several efforts to produce various information visualization taxonomies [Shneiderman96, Chi97, North98, CMS99]. In this paper, we will present a detailed analysis of a large number of visualization techniques using the Data State Model. The contribution is that our analysis of the information visualization design space is the most detailed and thorough to date. It is more detailed in the sense that we have broken each technique down by not only its data type, but also by its processing operating steps. It is thorough in that it categorizes the well-known techniques in [Chi97], [Olive99], and [CMS99].

2. RELATED WORK

Most previous work focused on constructing taxonomies of information visualization techniques uses a data-centric point of view. In an article describing the design space of information visualization techniques, Card and Mackinlay started constructing a data-oriented taxonomy [Card97], which is subsequently expanded in [CMS99]. This taxonomy divides the field of visualization into several subcategories: Scientific Visualization, GIS, Multi-dimensional Plots, Multi-dimensional Tables, Information Landscapes and Spaces, Node and Link, Trees, and Text Transforms. OLIVE is a taxonomy assembled by students in Shneiderman's information visualization class [Olive99], and divides information visualization techniques using eight visual data types: temporal, 1D, 2D, 3D, multi-D, Tree, Network, and Workspace.

Previously, we also proposed a taxonomy of information visualization techniques based not only on data types, but also on the processing operators that are inherent in each visualization technique [Chi98]. Elaborated in [Chi99], we showed that information visualization techniques could be described using the Information Visualization Data State Reference Model (or simply Data State Model).

3. SUMMARY OF DATA STATE MODEL

Figure 1 shows an overview of the Data State Model [Chi98], which breaks down each technique into four Data Stages, three types of Data Transformation and four types of Within Stage operators. The visualization data pipeline is broken into four distinct Data Stages: Value, Analytical Abstraction, Visualization Abstraction, and View (See Table 1). Transforming data from one stage to another requires one of the three types of Data Transformation operators: Data Transformation, Visualization Transformation, and Visual Mapping Transformation (Table 2).

Stage	Description
Value	The raw data.
Analytical Abstraction	Data about data, or information, a.k.a. meta-data.
Visualization Abstraction	Information that is visualizable on the screen using a visualization technique.
View	The end-product of the visualization mapping, where the user sees and interprets the picture presented to her.

Table 1: Data Stages in the Data State Model

Processing Step	Description
Data Transformation	Generates some form of analytical abstraction from the value (usually by extraction).
Visualization Transformation	Takes an analytical abstraction and further reduces it into some form of visualization abstraction, which is visualizable content.
Visual Mapping Transformation	Takes information that is in a visualizable format and presents a graphical view.

Table 2: Transformation Operators

Within each Data Stage, there are also operators that do not change the underlying data structures. These are the Within Stage Operators, of which there are four types, corresponding to the four Data Stages: Within Value, Within Analytical Abstraction, Within Visualization Abstraction, and Within View.

Figure 2 shows an example of the Data State Model applied to the problem of visualizing the connections between a set of Web pages. This example shows that: (1) some operators create new kinds of data sets, whereas some operators create filtered subsets, which is the difference between Transformation and Within Stage operators, and (2) that the same Visualization Abstractions can be mapped using a variety of Visual Mapping Transformation operators. For example, Disk Trees or Cone Trees can both be applied to a hierarchy of interconnected nodes.

4. TAXONOMY

By isolating dependencies, we can more easily reuse different parts of the pipeline to construct new information visualizations. Therefore, we have taken this model and used it to taxonomize various visualization techniques. The idea is to analyze the various techniques, thus increasing our knowledge of how each technique can be built using various operators. In the following, we used this model to analyze some 36 visualization techniques.

With a clearer understanding of the interactions between the data and the operators, implementers will be more equipped to construct new interactions or new visualizations. In practice, these analysis techniques have been applied in a system called the
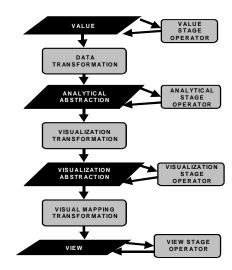

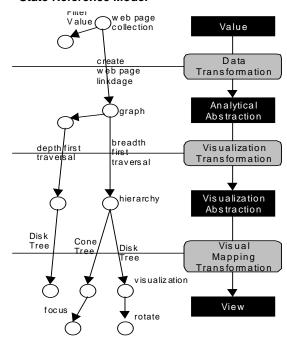**Figure 1: Information Visualization Data State Reference Model**

Figure 2: Data State Model applied to Web sites

Visualization Spreadsheet [Chi97, Chi99] and have enabled reuse and rapid development.

The following table presents the taxonomy using the Data State Model. A row represents a single visualization technique or system. The cells in that row describe the operators that comprises that technique. Non-italic items refer to the operators, while *Italic* items refer to example data sets within that Data Stage. The columns are the seven types of operators described in Figure 1: within and non-within stage operators.

In certain cases below, cells marked with → have no corresponding Abstractions or Operators at that stage, because the data is already in a visualizable format.

Visualization Technique	Within Value	Data Transformation	Within Analytical Abstraction	Visualization Transformation	Within Visualization Abstraction	Visual Mapping Transformation	Within View
Some example Scientific Visualizations							
Visible Human Project [NLM]	*Data:Image Scans of slices of human*	Marching Cubes: create voxels	*Abstraction: Voxels*	→	→	Create slices of Volumes	Specifying slices using sliders
MapQuest [MapQuest.com]	*Data: Geographical Road Maps*	Parse information into records	*Abstraction: Parsed record set* Dynamic value-filtering of records	Create linear list of records	*Visualization Abstraction: Linear list with item features*	Icons depicting different locations and their types: restaurants, etc.	Scroll, Zoom, View-filtering of interested locations.
Ozone Layer Visualization [Treinish94]	*Data: Ozone layer geographical information over time*	Extract geographical samples into quantitative variables	Normalize samples and quantitative values	Direct spatial mapping of quantitative values to longitude, latitude, height	*Abstraction: Earth with overlaid info.*	Map quantitative variables to longitude, latitude, and height; Map ozone level to color	Rotate, Scale, Animate; Change colormap
Geographical-based Info Visualization							
Profit Landscape [Visible Decisions]	*Data: Profit statistics linked to geographical regions*	Extract into quantitative variables	Normalize sample	Direct spatial mapping of geo-coordinate variables	→	Map geo-coordinate variables onto a geographical map; Map profit variable to glyph (size of lines)	Rotate, Scale, Animate; Change colormap
2D							
TileBars [Hearst95]	*Example data: text documents*	Parse into feature vectors.	Search through vector; compute intersection of vectors.	Each rectangle corresponds to a document.	→	Squares represent text segments; darkness indicates frequency of terms	Browse
ValueBars [Chimera92]	*Example data: text documents, file system records*	Parse into feature vectors, then choose one attribute.	Allow multiple attributes to be chosen for several ValueBars.	→	→	Lines represent the value of the attribute of an item in the text document.	Scroll
Information Mural [Jerding96]	*Example data: software code, documents, stock prices, sun spot data*	Parse into feature vectors	Dynamic Value-Filtering	→	→	Lines represent the value of the attribute of an item in the document; color maps another value or type.	Scroll; zoom
LifeLines [Plaisant96]	*Example data: medical and court records*	Parse into feature records	Dynamic Value-Filtering	Create lines on 2D plot	Dynamic value-filtering; Apply unmapped variable filtering	Icons indicate discrete events; Line colors and thickness indicate relation or significance	Dynamic view-filtering
Multi-dimensional Plots							
Dynamic Querying [Ahlberg94]	*Example data: Home, Movies sales data*	Parse into feature records	→	Create multi-dimensional point sets	Dynamic value-filtering; Apply unmapped variable filtering	Map into scatter plot; Choosing variables-to-axes mappings	Dynamic view-filtering
Parallel Coordinates [Inselberg97]	*Example data sets: production run of VLSI chip yield and its defect parameters*	Extract corresponding yield and parameter feature set	Choosing a subset of records using dynamic value-filtering	Create point set from records	*Visualization Abstraction: Point set*	Plot point set using parallel coordinates	Dynamic view-filtering; Sorting of axis; Interactive permutation of axis

Visualization Technique	Within Value	Data Transformation	Within Analytical Abstraction	Visualization Transformation	Within Visualiza-tion Abstraction	Visual Mapping Transformation	Within View
World-Within-World [Feiner 93]	*Data: High-dimensional point set or surfaces* Dynamic value-filtering	→	*Abstraction: Point set or surfaces.* Normalize samples	→	→	Map high dimensional surface to local area	Dynamic view-filtering; Rotate, Scale, Focus
Information Landscapes and Spaces							
Perspective Wall [Mackinlay91]	*Example Data: Schedule, File system*	Parse information into records	*Abstraction: Parsed record set* Dynamic value-filtering of records	Create linear list of records	*Visualization Abstraction: Linear list with item features*	Create wall panels in 3D with glyphs, with focus+context distortion-based display	Focus on a particular wall; Focus an item; Dynamic view-filter; Choose different levels of detail
Pad++ [Bederson94]	*Example data: Many types including text documents, file system, drawings.*	→	*Abstraction: windows, lines, icons, points, polygons.*	→	*Many Abstractions are compatible.*	Many representations.	Zoom: some objects will fade in, some will fade out Scroll
Elastic Windows [Kandogan96]	*Example Data: Mail reader, WWW Browser, Window based interfaces*	→	*Abstraction: windows*	→	*Many kinds of windows are compatible.*	Create Elastic Window mapping to screen (space-tiling)	Change focus, enlarge, zoom-out
WebBook and WebForager [Card96]	*Data: URLs for web pages*	Retrieve web pages; Generate images of each Web page	*Abstraction: Images of HTML pages generated by getting the Web pages*	Create linear list of pages; Aggregate into a book or a pile; Place pile on book shelf (creating list of lists); Crawl from a URL and create a book from the collection	*Abstraction: Linear page lists, Collection of page lists.* Merge page lists; Merge sets of page lists	Create books with multiple pages; View using Document Lens; Create bookshelf, table, piles	Focus on a book; Focus on a page; Flip through pages in a book; View book using Document Lens; Put onto history pile
Trees							
Cone tree [Robertson91], Hyperbolic Browser Lamping95], TreeMap [Johnson91], DiskTree [Chi98], Cheops [Beaudion96], WebTOC [Nation97], Information Cube [Rekimoto93]	*Data: File system; Organization charts; Hypertext or Web linkage structure*	Extract into graph	*Abstraction: Graph* Apply dynamic value-filtering of nodes or edges	Do breadth first traversal	*Visualization Abstraction: Tree hierarchy*	Layout using 3D cones; hyperbolic tree; Disk Tree; space filling TreeMap; Cheops approach; Expanding trees; Using Information Cube technique with semi-transparent cubes.	Focus node; Hide subtree; Change orientation and position of tree; Apply Dynamic level-filtering
Network							

Visualization Technique	Within Value	Data Transformation	Within Analytical Abstraction	Visualization Transformation	Within Visualization Abstraction	Visual Mapping Transformation	Within View
GraphViz [AT&T]	*Example data: Software code modules, File system, all kinds of graphs*	Extract edges and nodes into a graph.	*Abstraction: graph*	→	→	Sophisticated graph layout algorithm that places nodes on 2D plane intelligently	View: graph with minimized edge crossings.
GV3D or lpNV3D [Ware93]	*Example data: Software code modules*	Extract connections between code	filter nodes	Form nested graphs from earlier extracted graphs.	Dynamic value-filtering; Apply unmapped variable filtering	Map code modules into cubes in 3D, with linkages between cubes specifying relationships	Dynamic view-filtering
SeeNet [Becker95] Comment: aggregation is mentioned as implemented using data management software, several different views of the data sets.	*Example data sets: phone calls made; Internet packet flows; Email communication patterns*	Parse into source and destination links.	*Analytical abstraction: parsed records of source and destination and associated feature sets* Unmapped variable value-filtering; Choose variables of displayed statistics; Aggregate records	Transform into graphs and networks.	*Visualization Abstraction: Graphs, and Networks*	Display graph as matrix, geographical linkmaps, or nodemaps	For all three views: Sound feedback; Unmapped variable view-filtering (they called it 'conditioning') For matrix display: Threshold time view-slider; Permute rows and columns For nodemaps and linkmaps: Change Size, Color, Zoom; Parameter focusing; Identification by brushing; Change animation speed; Change line thickness, or line length; Dynamic query threshold view-slider For nodemaps: Change symbol size; Use color sensitivity view-slider
Text							
AlignmentViewer [Chi96]	*Data: Similarity reports from comparing a single sequence against a database of many other sequences*	Parsing textual reports; Addition, Subtraction between different reports; Unmapped variable value-filtering	*Abstraction: Alignment records (data structure representing parsed information)*	Extracting information from records	*Visualization Abstraction: Feature point set with vector*	Map into comb-glyphs	Rotation, Translate Zoom; Focus on a single alignment; Detail-on-demand; Animation (by using an iterator over the view-filtering)
ThemeScape and Galaxies [Wise95]	*Data: CNN news stories*	Create textual word frequency vector; Choose an item and then perform weighted query	*Analytical Abstraction: Text vectors*	Multi-dimensional scaling (MDS); Principal component analysis	*Visualization Abstraction: 2D positions from MDS*	Map into surfaces of hills and valleys	Zoom, Rotate; Focus on detail spot For ThemeScape: Create slices For Galaxies: Animate scatter plot
Web Visualization							
WebSpace [Chi94]	*Data: web site*	walk web site and create web linkage graph	value-filtering	Create breadth first traversal tree	*Visualization Abstraction: Tree*	Layout using cone tree	Dynamic view-filtering
3D Hyperbolic [Munzner95]	*Data: web site*	walk web site and create web linkage graph	value-filtering	Create breadth first traversal tree	*Visualization Abstraction: Tree*	Layout using 3D Hyperbolic Tree	Dynamic view-filtering
WebMap [Domel94]	*Data: web site traversal history*	Extract user path from traversal history graph	*Abstraction: traversal history graph*	Form navigation spanning trees.	*Visualization Abstraction: Tree*	Map to Tree layout, circle layout, rectangle layout, Horizon tree layout	Dynamic view-filtering

Visualization Technique	Within Value	Data Transformation	Within Analytical Abstraction	Visualization Transformation	Within Visualization Abstraction	Visual Mapping Transformation	Within View
Time Tube [Chi98]	*Data: web structure evolving over time and its associated usage statistics (Content, Usage, and Topology of the web site)*	Create graph from web structure by crawling the web site	*Analytical Abstraction: Evolving graph represented as ordered collection of graph*	Do breadth first traversal with global node position over time	*Visualization Abstraction: Evolving tree as ordered list of trees*	Create Time Tube, which is represented using an aggregation of Disk Trees (invisible tube--like shelf)	Recognize gestures for: Focus on a slice; Bring slices back into the Time Tube; Zooming focus on the connectivity of a node by right-clicking on it; Rotate slices; Brushing on pages by highlight URL on all slices; Animate through the slices
Visualization Spreadsheets							
Table Lens [Rao94a,Rao95]	*Data: baseball player statistics*	Parse statistics into numeric records	*Analytical Abstraction: Numeric records* Sort records	Construct numeric table from records	*Visualization Abstraction: Constructed numeric table*	Represent number using bars, with focus+context distortion-based table	Change distortion focus
Spreadsheet for Images [Levoy94] (direct mapping from data to view)	*Data, Analytical and Visualization Abstraction: pixels, voxels* Rotate Image; Filter; Change color scale; and other image processing mechanisms	→	→	→	→	→	*View: images from pixels, volumes from voxels* Rotate image; Filter; Change color scale; and other image processing operations; Rocking the volume visualization
FINESSE [Varshney96]	*Data: Financial data*	Compute call and put option prices	*Abstraction: Matrix records, Mathematical functions* Change parameter of functions; Change arithmetic relationships	Compute curves from math function models	*Visualization Abstraction: Matrix, Computed curves*	Create heat map; Create surfaces in 3D; Plot using 3D bar charts; 2D line plots; Create text for filenames; Represent variables using value sliders	Change orientation of geometric objects; Change to common colormap or font; View using same geometric orientation; Show cell dependency relationships; Picking a data item, Input math function
Spreadsheet for Information Visualization [Chi97infovis] Comment: allows value and view Dependencies between cells	*Example data sets: Point sets; Matrix; Sequence similarity reports; Web structure, Web usage pattern; etc.*	Normalize matrices; Parse textual reports; Create random point sets; Create graph from web structure by crawling the web site	*Abstraction: Normalized matrix and point sets; Value tuples; Evolving graph represented as ordered collection of graph* Dynamic value-filter; Algebraic data set operators	Perform Delaunay Triangulation; Extract data features from records; Do breadth first traversal with global node position over time	*Visualization Abstraction: Point set; Matrix; Triangulated surface; Point set with feature vector; hierarchy, list of trees; etc.*	Create heat map, matrix cube visualization, matrix bar visualization, Cone Tree, Disk Tree, glyphs, scatter plot; Choosing variable-to-axes mapping; Change cells to share same visual mapping Transformation	Dynamic view-filter; Change object position and orientation; Pixel image addition between cells; Geometric object addition between cells; Animation; Coordinated direct manipulation
Web Analysis Visualization Spreadsheet [Chi99]	*Date: Web site usage analysis* Filter-Value	Extract linkage information; Extract usage information	Cluster nodes	Breadth First Traversal	Perform usage frequency pattern algebra; Apply Spreading Activation pattern algebra	Display Disk Tree; Display Cone Tree; Apply Coloring Pattern; Display Pattern Glyph	Apply geometric operators; Detail-on-demand Zoom; Animation

5. DISCUSSION

The contribution of this taxonomy is that we have made another step toward understanding the design space of visualization techniques by extracting the crucial operating steps in each technique. Using example data domains for each technique, we described the operators that are possible. We also illustrated the power of the Data State Model by applying it to the design of many well-known visualization techniques. For both researchers and implementers, this analysis shows how each of these visualization techniques would be broken down and implemented in the Data State Model and how it could be used in a visualization system. We encourage the reader to examine the taxonomy in detail to explore the similarities between certain visualization techniques. We facilitate this analysis by grouping the techniques into several data domains.

For each of the visualization techniques, the results of the analysis help us classify and choose how to implement the different operators in a large visualization system. For example, many hierarchical techniques share similar operating steps that can be standardized in a system. Implementers may take advantage of these similarities. As another example, the taxonomy also points out that there are many systems that use graphs or multi-dimensional point sets as the primary Analytical Abstraction. Systems could be built to take advantage of this similarity, so that the techniques can be applied more broadly to many problem domains.

For implementers, the taxonomy also directly specifies the sequential ordering of operators that are possible in a given visualization technique. In this way, it specifies the system module dependencies that are induced between the operators. Knowing these dependencies enables implementers to better organize their system for modularity. For example, we have shown that this technique enables rapid development of new visualizations in the Visualization Spreadsheet system [Chi97]. Many techniques were implemented in hours rather than in days, because we realized the reusability of modules by identifying the module dependencies.

We chose the techniques based on their familiarity to the information visualization community and their relevance to information visualization systems. This set of techniques spans a range of visualization design space, as it is based on previous efforts on taxonomies of information visualization design space. In looking at each of the visualization techniques, we first determine the raw data, and how it is obtained in the system. We then construct the visualization pipeline according to the description of each of the techniques in the literature[1].

As the visualization field grows to include more new techniques, this taxonomy will obviously change. Future work will include taking the taxonomy and making more meta-analysis of the similarities and differences between the operators in different data domains. Given that the nature of the problem solving tasks, the Data State Model should remain a valuable tool in the analysis of the visualization design space. This is because the Data State Model helps categorization and taxonomization, which expose the dependencies between visualization modules and the similarities and differences among visualization techniques.

6. REFERENCES

[Card97] S. K. Card, J. D. Mackinlay. The Structure of the Information Visualization Design Space. Proceedings of IEEE Symposium on Information Visualization (InfoVis '97), Phoenix, Arizona, 92-99 Color Plate 125, 1997.

[CMS99] S. K. Card, J. D. Mackinlay and B. Shneiderman. Information Visualization: Using Vision to Think. Morgan-Kaufmann, San Francisco, California, 1999.

[Chi97] Ed H. Chi, P. Barry, J. Riedl and J. Konstan. A Spreadsheet Approach to Information Visualization. Symposium on Information Visualization (InfoVis '97), Phoenix, AZ: 17-24, 1997.

[Chi98] Ed H. Chi and J. T. Riedl. An Operator Interaction Framework for Visualization Systems. Symposium on Information Visualization (InfoVis '98), Research Triangle Park, North Carolina: 63-70, 1998.

[Chi99] Ed H. Chi. A Framework for Information Visualization Spreadsheets. Ph.D. Thesis, University of Minnesota, March 1999.

[Shneiderman96] Ben Shneiderman. The eyes have it: A task by data type taxonomy of information visualizations, Proc. IEEE Symposium on Visual Languages '96, IEEE, Los Alamos, CA (September 1996), pg 336-343.

[North98] Chris North. A Taxonomy of Information Visualization User-Interfaces. http://www.cs.umd.edu/~north/infoviz.html, 1998.

[Olive99] OLIVE: On-line Library of Information Visualization Environments. http://otal.umd.edu/Olive/, 1999.

---

[1] We have omitted the bibliography of all of the 36 techniques mentioned to save space. The interested reader should refer to [CMS99] or write to the author.